Winter 2015 www.TechWell.com

BETTER SOFTWARE™

A TECHWELL PUBLICATION

IS AGILE BREAKING PRODUCT MANAGEMENT?
You're not the only one who needs to be agile

QA IS A PROCESS
Why quality assurance is more than a department

MOVE YOUR TESTING FORWARD WITH TRAINING FROM SQE TRAINING

The more training you take the greater the savings!

Maximize the impact of your training by combining courses in the same location. Combine a full week of training for the largest discount!

Moving Testing Forward

2015 SPRING SCHEDULE

TESTING TRAINING WEEKS

March 23–27, 2015
Boston, MA

April 27–May 1, 2015
San Diego, CA

June 1–5, 2015
Chicago, IL

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Software Tester Certification—Foundation Level			Mastering Test Design	
Security Testing for Test Professionals		Integrating Test with a DevOps Approach		DevOps Test Integration Workshop
Fundamentals of Agile Certification—ICAgile		Mobile Application Testing		Mobile Test Automation Workshop
		Agile Tester Certification		Agile Test Automation—ICAgile
Essential Test Planning and Management		Measurement & Metrics for Test Managers	Leadership for Test Managers	Test Improvement for Agile
Risk-Driven Software Testing		Performance Load and Stress Testing		

Green background Indicates courses pre-approved for Project Management Institute PDUs.

SQE TRAINING
A TECHWELL COMPANY

For information on our 60+ Public and 40+ Live Virtual Course Dates visit www.sqetraining.com

TESTING TRAINING LEARNING OPTIONS:

Public

Instructor-led training in a city near you

LIVE VIRTUAL TRAINING

Live, instructor-led classes via your computer

eLearning

Self-paced learning, online

ON-SITE ADVANTAGE

Instructor-led training at your location

WAYS *to* SAVE

Take advantage of the different "Ways to Save" on training using our discount programs listed below. Purchase valuable software quality training for your whole team and save.

EARLY BIRD

Register 6 weeks prior to any training week course and receive 10% off per registered course day. Take a full week of training to save the maximum amount!

TRAINING WEEK — COMBINE AND SAVE

Combine specialized training courses in the same location and save. Discounts vary depending on the number of training days combined.

GROUP DISCOUNTS

Have a group and want to save more? Get details on our discount policy by contacting our Client Support Group.

ON-SITE ADVANTAGE

Bring any course to your location for team training. On-site training is both cost-effective and convenient for your team of six or more.

CONFERENCE — COMBINE AND SAVE

Save when you combine any of our pre-conference training courses with your conference registration.

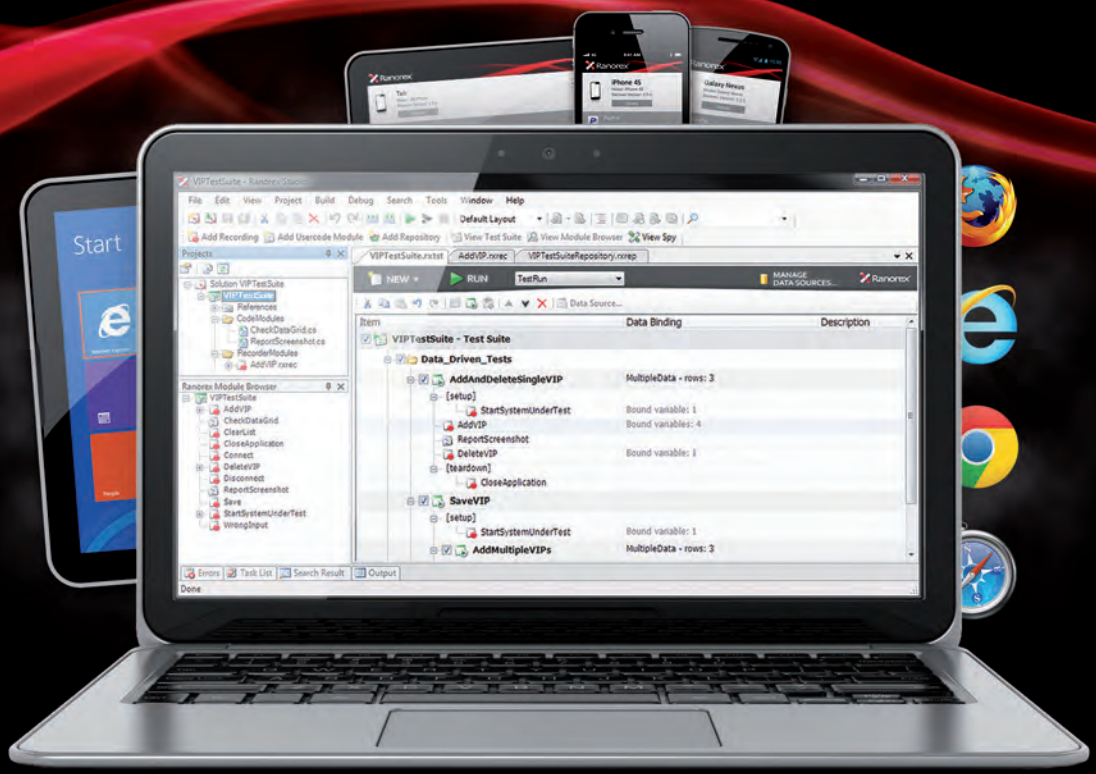
For more details on our discount policy, contact the Client Support Group at ***sqeinfo@sqe.com*** *or call 888.268.8770 or 904.278.0524.*

For information on our 60+ Public and 40+ Live Virtual Course Dates
visit www.sqetraining.com

SQE TRAINING
A TECHWELL COMPANY

Ranorex®

Automated Testing of Desktop. Web. Mobile.



- Robust Automation
- Broad Acceptance
- Seamless Integration
- Quick ROI

Why Use Ranorex
www.ranorex.com/why

Award-winning test automation tools provide seamless testing of a wide range of desktop, web and mobile applications.
Android | iOS | HTML5 | IE | FF | Chrome | Safari | WPF | Flash/Flex | Silverlight | Qt | SAP | .NET | MFC | Delphi | 3rd Party Controls | Java

BETTER SOFTWARE™
A TECHWELL PUBLICATION
Volume 17, Issue 1 • WINTER 2015

CONTENTS

in every issue

features

columns

MARK YOUR CALENDAR

SQE TRAINING

software tester certification

http://www.sqetraining.com/certification

Foundation-Level Certification
February 24–26, 2015
San Francisco, CA

March 2–4, 2015
Denver, CO

March 3–5, 2015
Charlotte, NC

March 10–12, 2015
Nashville, TN
Philadelphia, PA

March 16–18, 2015
Vienna, VA

March 17–19, 2015
Los Angeles, CA

March 23–25, 2015
Boston, MA

March 24–26, 2015
Toronto, ON

Advanced-Level Certification
February 23–27, 2015
San Francisco, CA

April 20–24, 2015
Atlanta, GA

training weeks

Testing Training Week
http://www.sqetraining.com/trainingweek
March 23–27, 2015
Boston, MA

April 27–May 1, 2015
San Diego, CA

June 1–5, 2015
Chicago, IL

Agile Training Week
http://www.sqetraining.com/agileweek
February 23–26, 2015
Atlanta, GA

May 11–14, 2015
Toronto, ON

conferences

Mobile Dev + Test Conference
http://mobiledevtest.techwell.com
April 12–17, 2015
San Diego, CA
Manchester Grand Hyatt

STAR*EAST*
http://stareast.techwell.com
May 3–8, 2015
Orlando, FL
Gaylord Palms Resort

Agile Development Conference West
http://adcwest.techwell.com
June 7–12, 2015
Las Vegas, NV
Caesars Palace

Better Software Conference West
http://bscwest.techwell.com
June 7–12, 2015
Las Vegas, NV
Caesars Palace

STAR*CANADA*
http://starcanada.techwell.com
June 21–25, 2015
Vancouver, BC, Canada
Westin Bayshore

STAR*WEST*
http://starwest.techwell.com
September 27–October 2, 2015
Anaheim, CA
Disneyland Hotel

BETTER™ SOFTWARE
A TECHWELL PUBLICATION

Publisher
Software Quality Engineering Inc.

President/CEO
Wayne Middleton

Director of Publishing
Heather Shanholtzer

Editorial

Better Software Editor
Ken Whitaker

Online Editors
Josiah Renaudin
Beth Romanik

Production Coordinator
Donna Handforth

Design

Creative Director
Catherine J. Clinger

Advertising

Sales Consultants
Daryll Paiva
Kim Trott

Sales Coordinator
Alex Dinney

Marketing

Marketing Manager
Kim Bryant

Marketing Assistant
Kelly Radell

CONTACT US
Editors: editors@bettersoftware.com
Subscriber Services:
info@bettersoftware.com
Phone: 904.278.0524, 888.268.8770
Fax: 904.278.4380
Address:
Better Software magazine
Software Quality Engineering, Inc.
340 Corporate Way, Suite 300
Orange Park, FL 32073

A Brand-New Year Brings New Changes

Happy New Year!

Just when we think there's nothing left for software developers to invent, we enter 2015 with more software innovation than ever before. Wearables just got a jolt in the arm with numerous smartwatches; standalone apps are giving way to app families that collaborate with most any device (phones, tablets, desktops, and the web); and Internet of Things devices are surrounding us with sensors and cloud connectivity.

Where teaching software development used to begin in high school or college, children are now being encouraged to learn how to code through the magic of innovative books, visual programming languages, coding camps, and videos.

But in the real world, just because you know how to code doesn't mean you know how to get past the hurdles to releasing a finished product to market. And that's exactly where TechWell and *Better Software* magazine come in. TechWell offers a wide range of training classes that will prepare you to better anticipate the complexity of issues plaguing anyone developing software. TechWell's conferences also offer tutorials and sessions to help you design software better, test better, and work with teams better.

Starting with this issue, *Better Software* will be published quarterly (it was bimonthly).

Our cover story by Brian Rabon gives you some great advice on becoming agile even if your company isn't. In a workshop I recently hosted at the Agile Development & Better Software Conference East in Orlando, I'd say 90 percent of the attendees stated that they worked for companies that claimed they were agile but really weren't. Talk about demoralizing. If quality is important to you (and it should be), you'll love reading Susan Bradley's article on treating QA as a process, not just a department. Steve Johnson, one the best product management consultants on the planet, wants you to know that product owners can be agile, too. And, because your team is supposed to thrive in a self-organizing environment, Maria Matarelli provides great advice about how to motivate your team throughout the entire product lifecycle. Lastly, if you're interested in creating an app on the side, you'll want to avoid the tough lessons Mike Botsko learned with his self-funded startup company.

Finally, our Technically Speaking column presents Matt Heusser's innovative approach that takes lean canvas concepts and applies them to building a test strategy. If you are in quality assurance, you'll want to read this (and don't forget to visit Matt's web page by clicking the StickyNotes link at the end of his column).

We truly value your feedback. Let us and our authors know what you think of the articles by leaving your comments. I sincerely hope you enjoy this issue.

Ken Whitaker

kwhitaker@sqe.com
Twitter: @Software_Maniac

Contributors

Mike Botsko is the owner of Helion3, a web and mobile application development company. Involved in web and application development since 1997, Mike has led development projects for companies large and small. Living in Oregon with his wife and two kids, he spends his free time fiddling with technology, games, and outdoor activities (kids and weather permitting). Contact Mike at botsko@gmail.com.

A certified software testing engineer, Susan Bradley has worked in quality assurance and software testing for eighteen years. Susan strives to bring the best of both manual testing processes and lean principles to her QA department. A writer of young adult mysteries in her off time, Susan has two published novels, *Unraveled* and *Uncovered*. You can follow her on Twitter @susanxbradley or visit her website, http://www.sxbradley.com.

The creator of the lean software delivery method, Matt Heusser is the managing consultant at Excelon Development, where he manages placement, consulting, training, subcontracts, and writing projects. Matt is a former board member for the Association for Software Testing and former part-time instructor in information systems for Calvin College. You can read his as editor work at StickyMinds.com and his columns as a contributing writer at CIO.com, or email him directly at matt@xndev.com.

Steve Johnson is a product management process coach, author, speaker, and advisor. His approach is based on the belief that minimal process and simple templates result in a nimble product marketing and management team. Steve has held positions from product manager to chief marketing officer with technology companies since 1981 in technical, sales, and marketing roles. At Under 10 Consulting, he guides product managers to be successful in an agile world. Contact Steve at steve@under10consulting.com.

Maria Matarelli is the founder and president of Formula Ink, a company dedicated to helping individuals and organizations reach unparalleled results. With more than ten industry certifications and as one of the few certified Scrum trainers, Maria and her team—with a specialty in enterprise agile transitions—coach executives and offer top-quality training for companies. With a passion for helping entrepreneurs, Maria provides rapid business development solutions for entrepreneurs to launch and grow sustainable businesses. You can reach Maria at maria@formulaink.com.

When not working on his theory of time travel, Cameron T. Philipp-Edmonds is writing for TechWell and its community sites. With a background in advertising and marketing, Cameron is partial to the ways that technology can enhance a company's brand equity. In his personal life, Cameron enjoys long walks on the beach, romantic dinners by candlelight, and playing practical jokes on his coworkers. Contact Cameron at cphilippedmonds@sqe.com.

Brian Rabon is the president of The Braintrust Consulting Group, a worldwide leader in agile transformations. Throughout his seventeen years of IT industry experience, Brian has applied agile methods to successfully deliver working products to his customers. When not in the classroom, Brian can be found around the globe evangelizing the benefits of agile to Agile 20xx, PMI chapters, and the Scrum Alliance's gatherings. Brian is the author of *Scrum for the Rest of Us!* and is an avid blogger. To contact Brian, email him at brian.rabon@braintrustgroup.com or join him on http://www.linkedin.com/in/bmrabon.

Michael Sowers has more than twenty-five years of practical experience as a quality and test leader of internationally distributed test teams across multiple industries. Michael is a senior consultant who works with large and small organizations to improve their software development, testing, and delivery approaches. He has worked with companies including Fidelity Investments, PepsiCo, FedEx, Southwest Airlines, Wells Fargo, and Lockheed to improve software quality, reduce time to market, and decrease costs. Reach Michael at msowers@sqe.com.

The Lean Test Canvas

The role of testing is often assumed to focus only on product validation. What you need is a strategy based on the lean test canvas.

by Matt Heusser | matt@xndev.com

Imagine for a moment that you run into your CEO in the parking lot. To your surprise, the CEO recognizes you by name and role, pointing out that the project you are working on is the company's biggest initiative for the year—perhaps for the next five years—and that keeping risks under control on this project is critical. She knows that you do something with "QA or something like that," and after an awkward moment, she asks, "What does your team do, exactly?"

How are you going to respond?

Visualizing the Model

The business model canvas is a visualization of a business model that can be represented on a single page of paper. [1, 2] Over the course of the past year, I have adopted this model for testers, using it to help them describe the value of testing and set expectations for a test group. I call it the lean test canvas.

We organized the canvas to start with the customer, move to what we add to the customer, then talk about testing: what the test process is and how it is accomplished (scope of role), then including all of testing, what is not included, and how the team can scale, improve, and demonstrate results. I have created a graphical representation of the lean test canvas at http://xndev.com/2014/11/the-lean-test-canvas.

> "The business model canvas is a visualization of a business model that can be represented on a single page of paper."

Start with the *scope of role* that includes all the events of the work and finish with key measures that defines how you'll know that all of the work has been successfully accomplished. [3] Here is a summary of each element.

1. ***Customers:*** Identify whom you serve. This can be developers, internal customers, the product owner, or the end-user. Order the list by priority.
2. ***Value proposition:*** Explain why the company employs testers as a role or testing as a service.
3. ***Test and deploy pipeline:*** What is the total calendar time elapsed from getting a new build to production deployment for planned work? This is typically stated as a range, like one to three days.
4. ***Core scope of role:*** Describe what your testers do. Nearly all teams test new features and perform regression tests, but do testers participate in product development? Security validation? Performance testing? Identifying product management risk?
5. ***Impact and mechanisms to monitor and rollback:*** State how quickly you'll be notified if something goes wrong in production. Most importantly, indicate how long it takes to roll back the last major change. Mention specific tools or techniques, such as feature flags or uptime graphs.
6. ***Key resources and activities:*** Specify how the team accomplishes its goals and key activities the testers perform every workday. This element can bring out activities that other team members did not even know were important, like checking log files, participating in building requirements or stories, or checking analytics tools to see how customers are using the system.
7. ***Out of scope:*** What does the team not do? For example, you might not perform security, performance, or unit testing. Knowing which activities are out of scope can be almost as important as knowing which activities are performed.
8. ***Cost structure and scaling:*** If upper management proclaims the team has to move faster, what resources or approaches could you employ to accelerate your work?
9. ***Direction of improvement:*** What is the goal, and what does improvement look like? Is it less downtime in production, shorter test or deployment cycles, releasing to production more often, processing more transactions in the same period of time, handling more users with performance degradation, or some other form of scaling?
10. ***Key measures:*** What metrics can you employ to know if your work is successful—or, at the very least, improved?

Don't think of filling in the boxes as a simple exercise that is performed once and forgotten. The real power of the lean test canvas is the conversations it forces the entire organization to have with the technical team members, management, and customers. These conversations ensure the team agrees on what it does, which brings focus and direction to delivery and improvement.

How to Use the Lean Test Canvas

Distribute copies of the lean test canvas to the team. Begin by discussing each element before instructing everyone to fill in the form. Once everyone understands what each element means, ask the team to fill out the form. A long-standing, high-functioning team might fill out the form together, but I'd suggest starting with each team member filling in the form individually. Pass the completed forms around for review, and encourage a lot of discussion as the group fills in the form together. The discussion should serve to unify the team, so this activity can be a great team-building exercise.

You may wish to consider asking management and your customers to fill in the form, too. A better comprehension of the testing role should create a shared understanding of what your testing team does to ensure quality is a key ingredient of any project.

I've created a full-page template you can use to create your own lean test canvas. [3] Post it in a highly visible place to serve as a reminder of your QA organization's focus. Periodically revisit the canvas to ask the team if the fields are still relevant, if improvements have taken place, and what to focus on next. **{end}**

Sticky Notes

Click here to read more at StickyMinds.com.

- References

Jeff Nielsen

Years in Industry: **14**
Email: **jeff.nielsen@3pillarglobal.com**

Interviewed by: **Cameron Philipp-Edmonds**
Email: **cphilippedmonds@sqe.com**

"For so many businesses (whether you're a software business or not), your software is your brand. It's the primary way that you interact with your customers."

"[In a larger enterprise] there are more people that want to be involved in decisions and probably more people that ought to be involved in decisions."

"Anyone these days that's involved in software in any way is foolish to ignore the lessons of the agile movement."

"I think that [the government] is ripe for agile. I know several leaders in the federal government personally, and you're going to see great things there [in the next few years] in terms of more of an agile approach and agile mindset."

"I think the agile mindset is basically a belief that the best software gets created in a collaborative, iterative, trial-and-error type of environment with lots of open communication and healthy interactions."

> "I expect it [agile] to become less of a thing and more of just a way that stuff gets done."

"You're not going to hit a day in your company where you say, 'Oh, congratulations, everyone now has an agile mindset.' You can't just hand someone a set of values and say, 'Start believing this now. This is our new approach.'"

"The way to structure a company today and the age of the customer is really to push power down to small teams and let them go."

STICKYMINDS.COM™

For the full interview, visit
https://well.tc/IWAE17-1

DEVELOP *and* TEST *for the* MOBILE FUTURE

SCANNING THE CONFERENCE PLAN

Choose from a full week of learning, networking, and more

SUNDAY
Multi-day Training Classes begin

MONDAY–TUESDAY
In-depth half- and full-day Tutorials

WEDNESDAY–THURSDAY
Keynotes, Concurrent Sessions, the Expo, Networking Events, and more

FRIDAY
Mobile Innovation and Leadership Summit

website now live at
MOBILEDEVTEST.TECHWELL.COM

CHOOSE *the* SESSIONS THAT MATTER MOST *to* YOU

MOBILE DEV + TEST

WEDNESDAY, APRIL 15

Mobile Development

W1 **Real-Time Contextual and Social Relevance in Mobile**
Jason Arbon, Applause

W4 **Mastering Android Development Tools**
Luke Wallace, Bottle Rocket

W7 **From Web Developer to Hybrid App Developer**
Greg Avola, Untapped

W10 **A DevOps Approach for Building 100 iOS Apps**
Leigh Williamson, IBM

Mobile Testing

W2 **Automate Your Way to Sanity in the Chaos of Mobile App Development**
Josh Anderson, Dude Solutions

W5 **Continuous Mobile Testing for Critical Business Apps**
Steven Winter, FIS Mobile

W8 **Mobile Test Automation with Big Data Analytics**
Tarun Bhatia, Microsoft

W11 **Mobile Testing Trends and Innovations**
Melissa Tondi, Denver Automation and Quality Engineering

Wearables/Smart Technology

W3 **Embed Yourself in Our Wearable Future**
Noble Ackerson, Byte an Atom Research

W6 **Wearing UX—When Our Clothes Become the Interface**
Jason Snook, CapTech Consulting

W9 **Using Apps to Help Users Experience Wearables and IoT Devices**
Chris Beauchamp, Crittercism

W12 **The Coming Avalanche of Wearable Mobile Apps**
Philip Lew, XBOSoft

THURSDAY, APRIL 16

Mobile Development

T1 **Today and Tomorrow: Mobile's Impact on Development and Testing**
James Montemagno, Xamarin

T4 **Plotting a Better Beer with Geolocation: An Untappd Case Study**
Greg Avola, Untapped

T7 **WIndows Azure: Connecting the Dots for a Mobile Workforce**
Mike Benkovich, Imagine Technologies, Inc.

T10 **Architect a Winning Mobile Application**
Shadi Saifan, FIS Mobile

Mobile Testing

T2 **A Look into the Future: App Testing and Quality in 2025**
Jason Arbon, Applause

T5 **Mobile Application Dev and QA Testing with Simulated Environments**
Wayne Ariola, Parasoft

T8 **Tips and Tricks for Building Secure Mobile Apps**
Jeff Payne, Coveros, Inc.

T11 **Mobile Performance Testing Crash Course**
Dustin Whittle, AppDynamics

Wearables/Smart Technology

T3 **TBA**
TBA

T6 **Wearables and Contactless Technology—for Payment Processingand Much More**
David Meyer, FIS Mobile

T9 **The Internet of Things and You**
Jim McKeeth, Embarcadero Technologies

T12 **Software Attacks for Embedded, Mobile, and Internet of Things**
Jon Hagar, Independent Consultant

WHO SHOULD ATTEND

Mobile Dev + Test is for designers, developers, and testers of mobile or smart applications. This includes product owners, software testers, test managers, QA managers and analysts, developers and engineers, mobile project managers and leads, and all managers, directors, CTOs, CIOs. Professionals already working with or that are interested in mobile, embedded or smart technologies.

TO REGISTER CALL 888.268.8770 | MOBILEDEVTEST.TECHWELL.COM

Featuring fresh news and insightful stories about topics that are important to you, TechWell.com is the place to go for what is happening in the software industry today. TechWell's passionate industry professionals curate new stories every weekday to keep you up to date on the latest in development, testing, business analysis, project management, agile, DevOps, and more. Here is a sample of some of the great content you'll find. Visit TechWell.com for the full stories and more!

Working with Your IT Talent to Support More Mobile Devices

Eric Bloom

Is your IT organization being pushed by your customers or partners to support every mobile device under the sun? While the support of multiple devices certainly has its business advantages, it also brings technical complexity, the need for knowledge of multiple operating systems, device-specific idiosyncrasies, and having to hire or train staff for the skills necessary to perform the required tasks.

If you can find them, you can hire new people or bring in a vendor to provide the needed support. If, however, you decide to use your existing staff for this purpose (for budgetary or other reasons), the question is who in your existing team has the most relevant and transferable skills to support a mobile environment with multiple devices.

https://well.tc/xh4

What Are the Top Cyber Threats for 2015?

Pamela Rentz

When Americans worry about crime, these days they worry more about their credit card information being hacked (69 percent) than they do about criminals breaking into their home (45 percent), stealing their car (42 percent), or even getting mugged (31 percent), according to a recent Gallup poll.

The Gallup annual crime survey also reported that during the past year, more than a quarter of Americans indicated that they or another member of their household had their information from a credit card they used at a store stolen.

https://well.tc/fYi

The Top Four Myths about Web Accessibility

Anish Krishnan

The Web Accessibility Initiative was launched in 1997, yet this many years later, it is still a widely ignored and neglected aspect of web development. There are many deep-rooted misconceptions about accessibility that prevent people from making a conscious effort to incorporate accessibility into their websites.

https://well.tc/fPb

The Project Manager-Business Analyst Relationship: When Roles Collide

Joy Beatty

I've recently been thinking about the relationship between a project manager (PM) and a business analyst (BA). In some organizations, these roles are played by different people, and in other organizations, one person performs both. I'll suggest tips for both scenarios.

My preference is that organizations separate who performs the project management role from who performs business analysis. The two roles require different skill sets, and it's hard to find one person who has the skills to do both jobs well. Secondly, I think it is a rare project that is small enough that one person actually could fill both roles sufficiently.

https://well.tc/xyb

Mind the Orchestration Gap in the Cloud

Beth Cohen

Do you find that the orchestration tools for building applications in the cloud seem to be missing key functionality? Orchestration tools hold great promise for making it easier to build cloud-based applications that are able to take advantage of the flexibility and cost savings of cloud infrastructures. Some new enterprising companies are working on building tools that integrate the application and network layers, which will allow developers to build cloud applications across hybrid environments faster and easier—and hopefully cheaper.

https://well.tc/YSu

Skills Needed to Better Leverage the Internet of Things

Cameron Philipp-Edmonds

In today's world, every device and machine seems to have embedded software or an internet connection capability associated with it. This phenomenon is the "Internet of Things," and it is ushering in a transition of greater opportunities for just about everyone.

But the transition into life with the ever-present Internet of Things (IoT) won't be a smooth transition for everyone. Luckily, Sarah Miller Caldicott, an industry leader in thought innovation, has narrowed down a list of five innovation skills that will help individuals and companies become better prepared for the IoT revolution. With Caldicott's list in mind, here are three skills that you'll need to better leverage an ever-connected society to your advantage.

https://well.tc/YGg

Maintaining Tester Independence in an Agile World

Michael Sowers

As organizations adopt agile methodologies, one of the key challenges is reinventing traditional roles such as project manager, functional manager, business analyst, developer, and tester. While many of the tasks and accountabilities for these

roles must still be carried out, they are set in a much different context: one of a homogenous agile product team.

Ideally, these responsibilities could be carried out by anyone on the agile product team, but in practice, each team member will have his or her own individual skills, experiences, and strengths. One primary shift is that the entire agile team is accountable for quality—carrying the quality flag is not the sole responsibility of the tester.

https://well.tc/Ycd

Analyzing the Value of a Test Tool Approach

Matt Heusser

Over the past ten years, questions I get from test managers have subtly shifted from "How can we automate our process?" to "How do we assess the value of our test tool approach?" Hidden within that second question is a fear that the effort lacks the return senior management was hoping for.

There is a way to analyze the value of a test tool approach that does not require writing code—only the ability to read it a little.

https://well.tc/YmW

In Software Development, Execution Isn't Everything

Joel Basgall

I'm sure you hear some version of this statement all the time: "Ideas are cheap. Execution is everything."

This is true, but only to a point. Where software development is concerned, good and even great execution alone does not ensure a successful product.

While there certainly are best practices, it's also important to embrace the fact that every project, every product, and every idea is different. Although you can sometimes leverage the past and apply what has worked before, often you need to dream up something new. That means execution comes later.

https://well.tc/xhz

Maximizing the Coexistence of Scrum and Kanban

Manoj Khanna

It is difficult to exclusively use Scrum or kanban in product development, given the advantages they both provide. The prospect of using the two together can be just as difficult to fathom, yet it is possible for them to coexist —and with optimal results.

First, let's compare and contrast the two terms. Scrum is an agile framework used to manage and complete projects, whereas kanban is a nonagile technique used to direct development processes. Scrum is used to organize the project as a whole, from team roles to what composes a product's development. Those involved in product development include the product owner, ScrumMaster, and development team members who work to tackle and complete the necessary tasks in each sprint.

https://well.tc/N6b

Early Testing Questions for Mobile Apps

Jon Hagar

Assuming one has considered test strategy and done some high-level test verification and validation (V&V) planning, a mobile app tester should take early actions to gain a better understanding of the product in order to answer the following questions:

- What test/V&V actions has the developer taken?
- Has the developer implemented static code-analysis testing?
- What risks exist for the product?
- Does the team communicate well and understand what software development/test actions are in play?

https://well.tc/Nho

Why Collaborative Workspaces Are Not for Everyone

Naomi Karten

What comes to mind when you think of collaborative workspaces? My own reaction is noise, interruptions, lack of privacy, inability to concentrate, and being in each other's faces. But maybe I'm looking at it all wrong; after all, cubicles don't exactly facilitate collaboration.

EF Education First, a company on the forefront of collaborative workspaces, offers conference rooms with glass walls looking out onto panoramic views, and large, colorful living room-like spaces that serve as meeting areas. That sounds nice, especially given research that has found an association between lack of windows and decreased productivity. It seems the absence of natural light affects the ability to get to sleep at night, and less sleep translates into reduced productivity.

https://well.tc/Yib

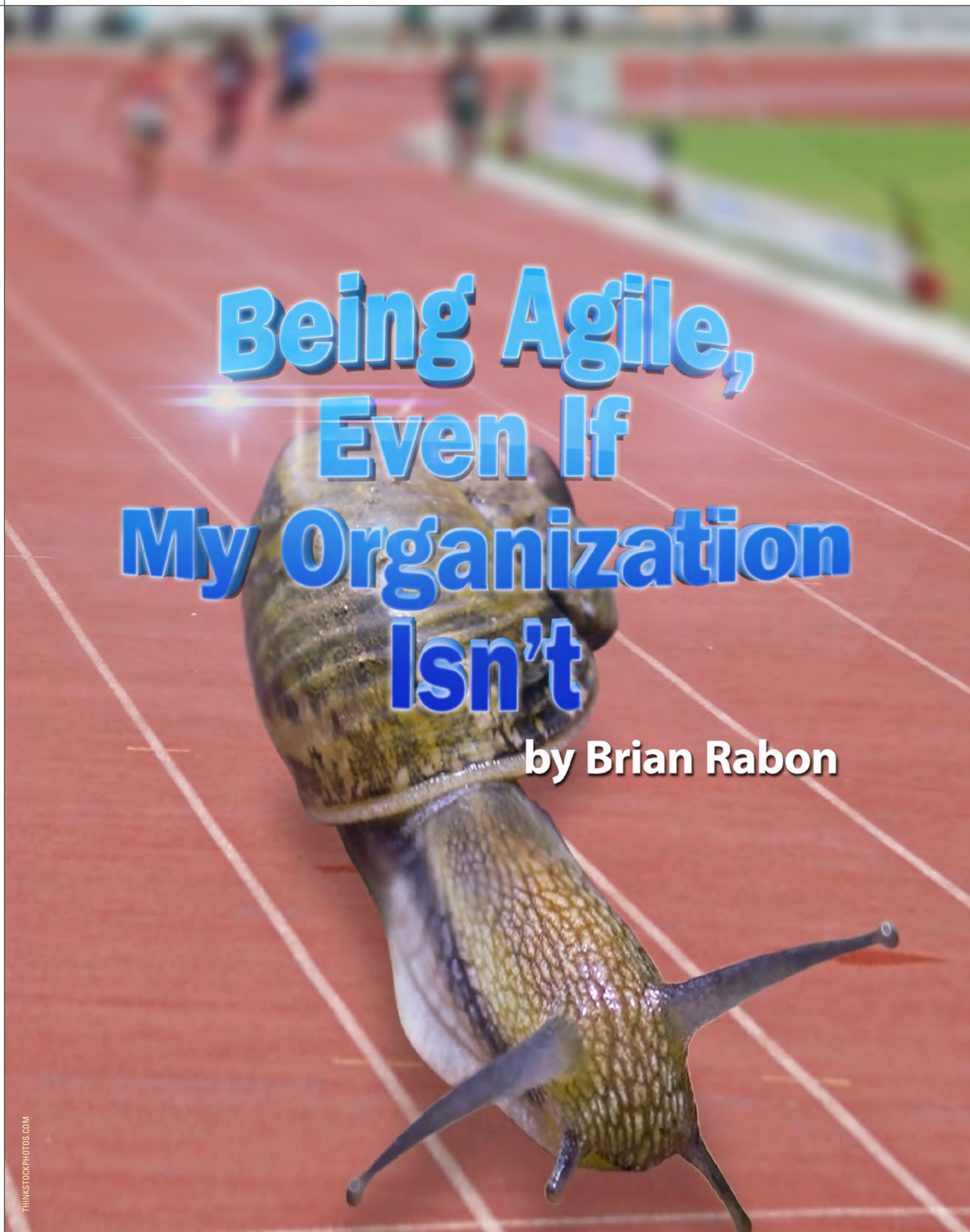
Performance Testing: A Team Sport

Dale Perry

Performance testing is fundamentally different from functional testing. In functional testing you design test cases to address specific issues and risk inherent to the product being tested, focusing on both positive (does it work?) and negative (locating defects) aspects of the product.

When executing a test case you have an expected result in mind, and each time you run the test, you expect the same result. Functional testing is about what the system does.

https://well.tc/Yad



Being Agile, Even If My Organization Isn't

by Brian Rabon

THINKSTOCKPHOTOS.COM

For the majority of software development professionals, a passion for lifelong learning is ingrained in our souls. What have you done to personally embrace agility?

- Are you still waiting for your job to provide training?
- Are you trying to go through the motions at work but don't feel technically fulfilled?
- Are you ignoring or resisting agile because you are hoping it's a fad and will pass?

If any of these statements apply to you, then perhaps it's time to step up and take personal responsibility for your own growth and learning. Does the following quote resonate with you?

> *"Are you green and growing or ripe and rotting?"*
> *—Ray Kroc*

Agile is here to stay, and it's quickly becoming the best way software development is being done. When one of Scrum's most influential founders, Mike Cohn, was asked to comment on the future of agile, he said, "I hope we've stopped calling it anything at all and are just doing it." [1]

This article examines how we can personally become more agile, regardless of what is going on with our organizations. You will also learn how being more agile can help you survive in today's world and prosper in the future.

What Is Agile, Anyway?

Many people mistakenly think agile is a methodology or a process, when in fact you should think of agile as a philosophy or a mindset. When the founding fathers of the agile movement got together on a snowy February 21, 2001, at the Snowbird Ski and Summer Resort in Utah, they immortalized their vision consisting of four values and twelve principles as the basic tenets of the Agile Manifesto. [2]

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

> "Individuals and interactions over processes and tools
> Working software over comprehensive documentation
> Customer collaboration over contract negotiation
> Responding to change over following a plan"

While there is value in the items on the right, we value the items on the left more.

This agile philosophy is deeply ingrained in processes for getting work done, like Scrum and Extreme Programming. The trend in our industry has evolved to where most of us are now developing software as Scrum teams.

Unfortunately for many of you, your teams may be going through the motions of Scrum, but your organizations may not be actually living the agile values and principles. Scrum becomes a soulless process that you follow robotically because it is what is expected of you.

How Agile Is Your Organization?

How do you know if your Scrum implementation is lacking soul?

- Does the daily scrum feel like a status meeting for the ScrumMaster?
- Are your retrospectives getting shorter and shorter—and providing less and less value?
- Does management still hold you accountable for producing traditional planning documents like Gantt charts and providing reports on metrics like earned value?

Answering "yes" to these questions is a sign that your organization has adopted the mechanics of the process but not the agile culture. So, what is agile culture, and how to do we influence it?

In order to begin to adopt an agile culture, we need to start looking at our own personal values that define our nature. Behaving in an agile way requires living the values of personal agility, detailed in what The Braintrust Consulting Group calls the Manifesto for Personal Agility shown in figure 1. [3]

Manifesto for Personal Agility

We are uncovering better ways of living our lives by doing it and helping others do it. Through this work we have come to value:

- Setting a vision for our life over letting fate have its way
- Partnering with others over trying to do it all ourselves
- Getting things done over wasting away our days
- Looking internally over blaming others

That is, while many of us live on the right, we choose to live on the left.

Living the Agile Life

In order to begin living the values and principles of agility,

Figure 1: The seven facets to achieving personal agility

you need concrete actions that you can take. Actions, in terms of developing skills in seven different domains, will give you a framework for developing your own competency at becoming more agile. Consider these seven domains as the syntax for your own personal transformation. Once you master the syntax and the skills, you should be able to solve any problem that you encounter in life.

Think of the seven domains as the facets of a diamond. When all of them are mastered successfully, the diamond will shine brilliantly. When one or more area is underdeveloped, the diamond will appear dull. We need mastery in all seven domains in order to shine and be agile.

"You" are at the center of the diamond because the foundation of personal agility is your values. Have you taken the time to reflect on what you stand for? Having a strong set of personal values will ensure that you are able to reflect on the rest of your life with meaning and purpose.

1. ***Looking inward:*** Many of us will hit plateaus in our lives, areas where we feel that we are stuck and not growing. When this occurs to you, recognize it for what it truly is—an opportunity to stop and look inside ourselves at what we need to do to take our lives to the next level.
2. ***Clear vision:*** We ask children, "What do you want to be when you grow up?" Once we become adults, we stop asking this question. Have you taken the time to dream like a child? Do you know what you want out of life? Life is too short not to go after your dreams.
3. ***Taking smart risks:*** Let's face it—we take risks every day. Some risks, like flying in an airplane, are more manageable than riskier endeavors like skydiving. When we get complacent in our lives, we stop taking risks because we are afraid of rocking the boat. In order to keep growing, we need get comfortable accepting calculated risks.
4. ***Entrepreneurial spirit:*** There are several key traits of an entrepreneur that can benefit us in our lives. Entrepreneurs take ownership of their work, entrepreneurs wear many hats, and entrepreneurs do whatever it takes to succeed. To be agile, we need to adopt the spirit of an entrepreneur.
5. ***Meaningful relationships:*** How many times have you thought, "I don't need other people, I can do it all myself." Next thing you know, you become all stressed out and overworked. By surrounding ourselves with others, you can form a team to help accomplish work and to be more successful in everyone's lives.
6. ***Situational leadership:*** Many of us think that we need to be a leader or a follower—and pick one. In reality, our situation may require us to decide on the fly. By recognizing when to step up and lead or to let it go and follow, we can be more successful at being agile.
7. ***Life as flow:*** Ever feel completely unmotivated to work? Do you tell yourself, "I will knock out that code when I have more energy"? Every day, our bodies go through cycles of motivation versus lack of motivation. If we can find ways to accomplish work at the times we are most motivated, we can be more productive.

Applying Personal Agility to Your Work

Remember that life is about the journey and not the destination. By taking the first step to improving yourself, you can start down a new path to achieving personal agility while future proofing your career. Let's examine specific domain practices that you can employ to achieve your own personal agility.

Looking Inward

In Scrum, the team holds a retrospective at the end of every sprint. During a retrospective, questions such as "What worked well this sprint?" and "What do we need to do differently next sprint?" are asked. The point of these questions is to give the team encouragement and to look for opportunities for improvement. Do you ever find that you need a pat on the back and some ideas to improve your software development skills? If so, then why not conduct frequent retrospectives on yourself? [4]

1. Choose a set of questions to ask yourself, such as: "What am I doing to continually market my skills and success? Is it working well? How am I viewed by my team? Do they see me as a mentor and key contributor to the team? If not, what could I be doing to change their opinion of me? Am I doing enough to keep myself current in new technology trends and ways of working? If not, what else could I be doing?"
2. Answer the questions. I like to put one answer per Post-It note in order to make the next steps easier. If you are concerned that you may not be objective, ask a friend or coworker for another perspective. Place the Post-It notes on the wall.
3. Group similar ideas and look for any patterns that emerge.
4. Prioritize the ideas you came up with. What is the one most important thing you can do right now? By selecting one idea to implement, you isolate a single variable and refrain from overwhelming yourself with too much change at one time.
5. Take the highest priority item and create a plan with assigned tasks and due dates in order to make it happen.

Meaningful Relationships

In his book *How the Best Get Better* management consultant Dan Sullivan describes a pattern of behavior he has coined "the rugged individualist." [5] rugged individualists believe he can do everything by himself. He doesn't need anyone else to help him.

It can be a solitary and lonely life. A software developer behaving as a rugged individualist may struggle on coding issues for days because he is too afraid to let others know that he needs help from others. He may isolate himself from his team because he doesn't want anyone to learn the real truth about his lack of skills or abilities.

We need to embrace what Dan calls the *unique-ability teamwork*. It is based on creative ways to get things done by determining each of our strengths and weaknesses. Rather than trying to improve your weaknesses, focus on your strengths by surrounding yourself with teammates who have strengths in your weak areas.

For example, I am getting ready to renovate a room in my basement. I am pretty confident that I could eventually learn how to do the electrical work, but by the time I figure it out, would it have been less expensive (and less frustrating) to hire an electrician instead? By forming my own teams in other areas of my life, I have been able to find the time to focus on my strengths rather than being bogged down doing things that I'm frankly not good at.

Make a list of all the things you are not good at, don't have time to do, or just plain don't enjoy doing. Ask your team members to do the same. Trade your weaknesses with a teammate who has that strength, and vice versa.

By surrounding yourself with teammates who can shore up your weaknesses, you will have more time to focus on your strengths. As you focus on your strengths, you should see rapid personal growth. In addition, you should feel more passionate about your work because you are only working in your unique ability.

Life as Flow

When I am working in the nine-to-five world, it feels like a prison sentence. I have to ask the warden (my boss) for permission to go to the doctor or to go on vacation. Consequently, when I am out of the office, I feel guilty and worried. What did other people think of my taking time off?

As we flow from one activity to the next in a given day, we are productive at times and then need some time to rest, relax, and rejuvenate. Spend a few hours mapping out your day. A simple system would be to put down either a happy face or a sad face for each hour of the day. A happy face means you are doing productive work; a sad face means you are not.

Employers are starting to wise up to the fact that not everyone is productive in the same ways at the same times. They are offering flex time, part-time telecommuting, or the option to work completely virtually. You may not be lucky enough to work for one of these enlightened organizations, so it may be time to put on your sales hat. Can you convince your boss to allow you to work remotely one day per week?

It's a start. When you control your day, you can live your life in the most productive flow state possible. You might actually spend less time coding, but in the end it could be even more productive.

Your Next Step to Personal Agility

Regardless of whether agile is being adopted at your workplace, you can become more agile yourself. By conducting frequent personal retrospectives you can fuel your personal and career growth. You can form meaningful relationships with coworkers in order to focus on your strengths and shore up your weaknesses. And you can learn to work at the right times in order to be most productive.

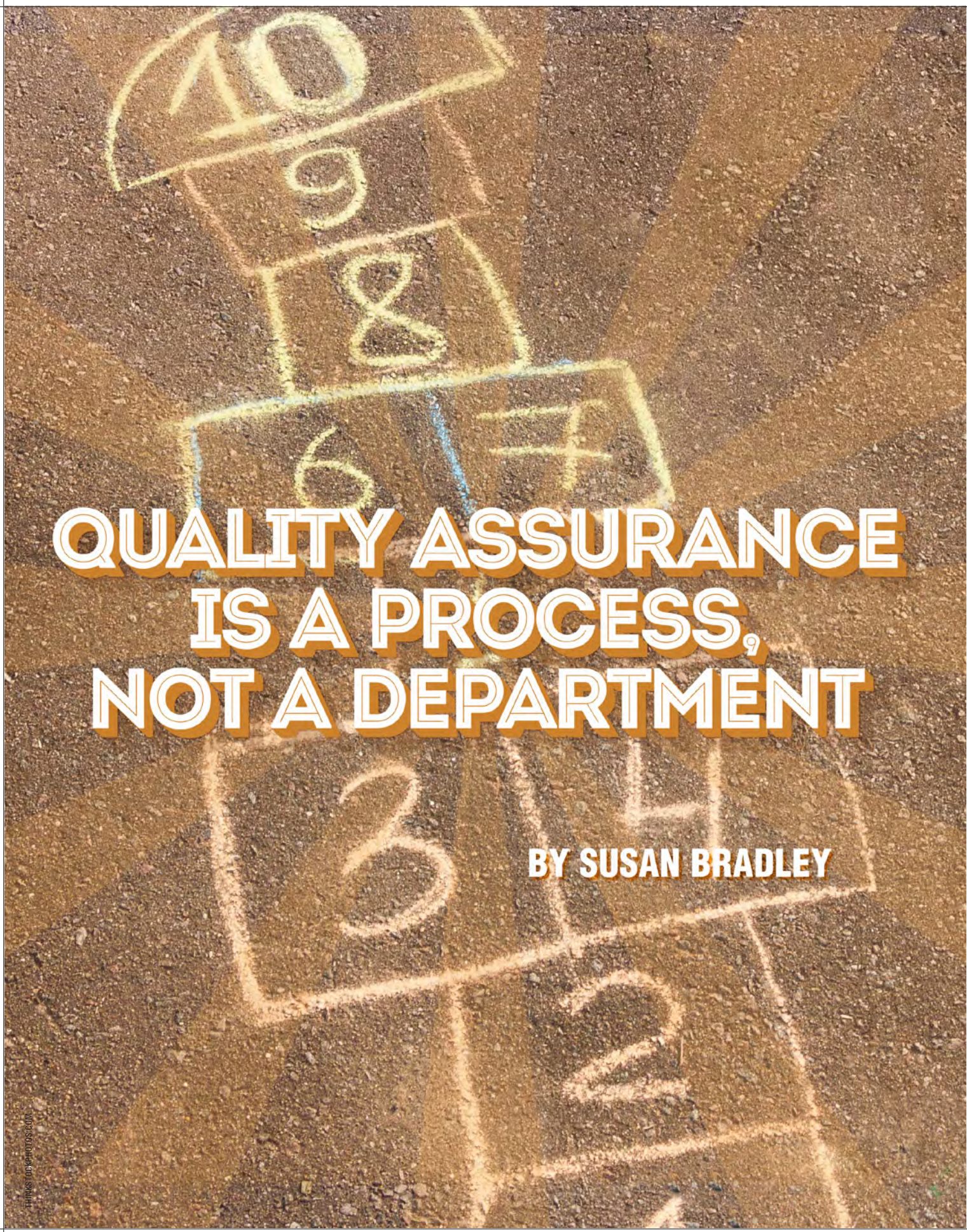
All of these techniques ensure that you are better suited to work in modern time. And the techniques should free up more time to focus on what is coming in the future. **{end}**

brian.rabon@braintrustgroup.com

Sticky Notes

Click here to read more at StickyMinds.com.

- References



QUALITY ASSURANCE IS A PROCESS, NOT A DEPARTMENT

BY SUSAN BRADLEY

THINKSTOCKPHOTOS.COM

All of us who work in quality assurance (QA) know the feeling of receiving that dreaded email or phone call: Something critical is broken in the production environment and customers are complaining. The first thing everyone wants to know is, "Was it tested?" We spend the next couple of hours with our hair on fire and a pit the size of a melon in our stomachs. We study our test cases, praying we did test the scenario that has produced the critical defect found by our customers. Here we are, once again, in the hot seat being asked to defend our work and our value to the organization.

As if the finger-pointing weren't bad enough, there is something everyone forgets when a severity 1 or critical issue occurs: the QA department never introduces bugs. QA does not write the requirements or the code. QA is an organization that is supposed to miraculously catch every single defect in every application as a sieve at the end of the software development lifecycle. We're supposed to exercise our x-ray vision to find the one bug lurking among millions of lines of code. After all, isn't that why a QA department exists?

Quality Assurance Is Much More Than Just Testing

After the drama is over and the critical issue has been mitigated, we may settle back and breathe a little easier. But the constant cycle of drama can wear down a QA analyst, leading to job dissatisfaction, burnout, and low morale. On a daily basis, QA manages a lot of moving parts, including tight deadlines, scope creep, requirement changes, and testing defects. In addition, we manage our test cases, testing execution, customer advocacy, defect resolution, and application expertise. Despite our knowledge of both the business and testing, we are expected to catch every piece of defective code in the source.

It's hard not to imagine Lucy and Ethel standing at the conveyor belt grabbing every piece of chocolate, trying to wrap it before the candy falls to the floor. Testing can feel just like that—we're running on all cylinders trying to deliver a quality product to the customer. You've probably heard the adage "You can't test quality into a software project; it must be built in."

As testers, we should be the final quality check before the product is released to the consumer. However, QA should not be the only quality check in the process.

How QA Fits into the Process

The basic process of a software release starts when the team receives a request from the business for a product change or a new piece of functionality. That request is converted into requirements or user stories by either the product owner or the business analyst. Those requirements are passed on to the development team to code, then finally on to QA for testing. There are opportunities for quality infusion in each of these steps. The National Institute of Standard Technology [1] published the cost of fixing a software defect in each key phase of the development process. This is shown in figure 1.

Not only is it more cost-effective to find defects early in a software development phase, but it is also much more efficient due to a significant reduction in rework. Although not shown in figure 1, defects can be introduced as early as the requirements phase that typically precedes design. The product owner (or business analyst) must interpret the business sponsor's vision and intentions in a way that will make sense to everyone on the project team. Much like creating blueprints for a home, this document should reflect what the end-user wants. Would you move forward with building a home without reviewing the blueprints first? I hope not. It's no different when creating software.

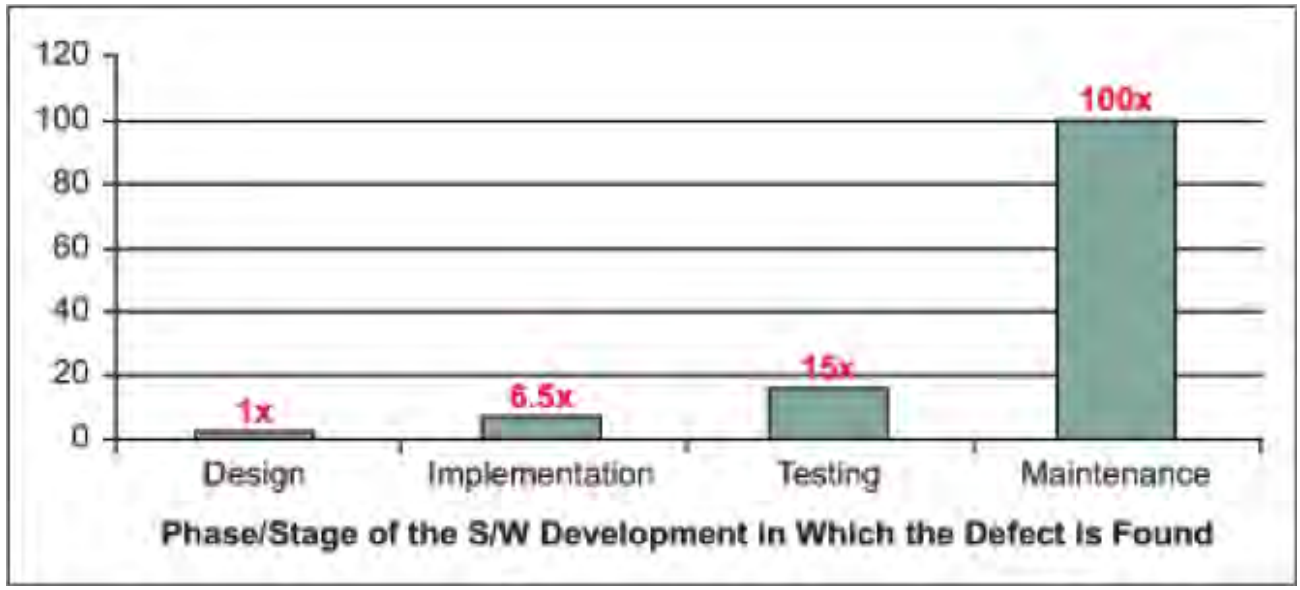


Figure 1: Relative costs to fix software defects (the vertical axis is a multiplier)

Reviews (or stage gates) at each phase of the lifecycle can remove defects with minimal effort and cost. Nothing has been coded or created, so there is nothing to take apart, tear down, or remove. My own perspective can be summarized as follows:

Embracing the QA process throughout the entire software development lifecycle helps to assure reaching business goals with:

- Faster speed to market
- Less expense and less risk
- Higher customer and employee satisfaction

W. Edwards Deming stated, "Quality is everyone's responsibility." [2] So, how do we begin to educate everyone involved throughout the lifecycle so everyone comes to think of QA as a process, not a department? This is the issue that many of us working in QA face.

What You Can Do

Promote teamwork: Attempting to convince technologists to work as a team can be a feat in itself, so fostering teamwork across multiple teams may seem impossible to achieve. One of the easiest, most effective ways to promote teamwork is to create an organizational culture of ownership among all software development lifecycle partners. This comes from the top and trickles down.

It is vital that your executives also endorse Deming's philosophy of the importance of QA throughout their organization. These leaders must inspire their management teams to participate in quality with all stakeholders. The managers, in turn, should spread that philosophy to their teams. Consequently, everyone will want to play a role in improving quality.

Educate and inform: It is one thing to ask people to do something because you want them to. It is another to get buy-in and support because your partners see the benefits of working together to improve the overall process. A monthly review of system test and production defects with all participating partners can be effective in a variety of ways. A root cause analysis can pinpoint specific areas that are opportunities for improvements, which can reside in the project management office, development, or QA areas. You never want to point fingers or place blame; instead, you should educate with facts. The purpose is to question why the problem occurred with the intent of preventing it from happening again.

Be on the lookout for defect reduction: Defects pose several risks to a company: diminished profit margins, employee downtime due to rework, loss of the company's reputation, and possibly litigation. Every production defect is a risk to a company. All employees must do their parts to mitigate risk by improving their commitment to quality.

Employ peer-to-peer collaboration: Insisting on transparency and cooperation can set the tone before a project or initiative begins so that everyone has the same understanding of the importance of quality. Leads from each participating department should meet and commit to injecting quality into each phase of the development process. Participants can brainstorm and determine which methods and tools can be used on a project, such as a variety of stage gates, peer reviews, and testing methodologies.

Why Stage Gates Should Be Used

Stage gates help ensure that the quality measures put in place have been met for each phase by providing a checkpoint to ensure that the project is ready to be continued. As an example, table 1 shows how stage gates are created during the planning phase of the project before a requirement, line of code, or test case is written.

The time spent up front to add these quality measures will have a return on investment of tenfold, a hundredfold, or a thousandfold, depending on which phase prevented the defect.

Why Wait? QA Starts during Planning

Moving quality to the left requires a change to a company's culture. Quality can begin before a word of requirements or a line of code is written.

The initial planning phase is an opportunity to start evangelizing the spirit of quality as a process. As the plan begins to take shape, the project team can see that the quality assurance team members are not the only ones responsible for quality. Quality is a team effort.

When trying out this new process, it is important to gather metrics along the way. They help support this switch from everyone's mentality of a QA department to a QA process. Records should be kept of how many defects are found during each phase.

There are some simple ways to determine that you're turning the corner. For example, each defect found prior to the testing phase is a savings to the company, both in terms of dollars and time. Rework is greatly reduced, project satisfaction is increased, and the pressure is lightened from the QA analysts. The goal of promoting quality across all departments should result in improved customer satisfaction by producing a quality product. Another benefit is that an emphasis in quality should build team collaboration and accountability. That's a win-win

	Quality	Status	Notes/Actions Required for Pass
2.9	Have project leads agreed on quality measures for each phase? For each department?		
2.10	Has a risk management assessment been conducted at the appropriate level based on the size and complexity of the project? Have mitigation plans been defined for the highest risk items?		
2.11	Have testing resources been engaged to determine an overall testing approach? Have any risks or issues identified with testing been incorporated into the plan?		

Table 1: Examples of stage gates

for any company.

In a perfect world, everyone would embrace this new way of approaching software development, but in reality, change can be difficult. How do you get the naysayers on board? You may not be able to convince everyone of the value of inserting quality versus testing it in. All you need is that one believer (and risk taker) who is willing to give the team approach to quality a chance with a beta project. That could be your opening to bringing this new way of developing software into the spotlight. Once quality improvements occur, it is important to promote the wins to everyone on the team and to upper management. Again, this is where metrics can be your best friend.

Utilizing Quality Metrics

If you can show the number of defects found by each quality check process, and then assign a dollar value in comparison to finding them in later phases, that cost savings can be a powerful number. Barry Boehm, software engineer and founding director of the Center for Systems and Software Engineering, said, "Finding and fixing a software problem after delivery is often a hundred times more expensive than finding and fixing it during the requirements and design phase." [3] This is consistent with the findings shown in figure 1.

A baseline metric based on the cost of production defects can be added to the defects found per phase metric. This baseline can be the cost of production defects per month, quarter, or iteration, depending on what makes sense for your place of work. Compare this against the savings of defects found prior to release. Those are quantifiable dollars—as savings toward a company's profit margin.

For example, let's say for the month of July there were fifty-seven production defects. For the sake of simplicity, let's say they were all injected during the requirements phase. Based on Boehm's rule of ten, the assumed cost for those defects is approximately $57,000. This does not include lost revenue or damage to a company's reputation.

How was that cost calculated? A root cause analysis can be conducted for each defect to determine which phase would have caught that defect. Based on Boehm's assumption, the following cost/defect breakdown can be used for each phase:

$1: Requirements
$10: Coding
$100: Testing

Add the number of defects that could have been caught during the requirements phase, then multiply that number by a dollar. In this example, twelve defects were found in the requirements phase. Do the same for those defects assigned to coding and testing. Add up the final cost and compare it against the $57,000.

Your math should look something like this:

12 ($1) + 30 ($10) + 15 ($100) = $12 + $300 + $1,500 = $1,812

The figure of $1,812 versus $57,000 should speak to any executive—and that's just for one month!

Going Forward

Armed with some baseline metrics, a plan, willing parties, and communication, any software business can change from viewing quality as a QA responsibility to accepting it as an interdepartmental process. Getting to quality inserted into each lifecycle phase won't happen overnight, but over time, the positive results should become tangible and quantifiable. Gradually, quality can become an integral process for any software development organization. **{end}**

bradley_susan@yahoo.com

Sticky Notes

Click here to read more at StickyMinds.com.
- References

STAR EAST

the QUEST *to* TEST

MAY 3–8, 2015
ORLANDO, FLORIDA
GAYLORD PALMS RESORT

REGISTER BY
MARCH 6, 2015
AND SAVE UP TO $400
GROUPS OF 3+ SAVE EVEN MORE

STAREAST.TECHWELL.COM

CONFERENCE EXPLORATION PLAN

Choose from a full week of learning, networking, and more

SUNDAY Multi-day Training Classes begin

MONDAY–TUESDAY In-depth half- and full-day Tutorials

WEDNESDAY–THURSDAY Keynotes, Concurrent Sessions, the Expo, Networking Events, and more

FRIDAY Testing & Quality Leadership Summit and Workshop on Regulated Software Testing (WREST)

OUT *of* THIS WORLD KEYNOTES

STAR EAST

Innovation: From the Tester's Viewpoint
Jason Arbon, Applause

Deliberate Testing in an Agile World
Dan North, Dan North & Associates

Lightning Strikes the Keynotes
Lee Copeland, Software Quality Engineering

Blunders in Test Automation
Dorothy Graham, Independent Consultant

The Future of the Software Testing Profession
Mike Sowers, Software Quality Engineering

JUST A FEW OF OUR IN-DEPTH HALF- AND FULL-DAY TUTORIALS

Rapid Software Testing: Strategy
James Bach, Satisfice, Inc.

Requirements Engineering for Testers
Erik van Veenendaal, Improve Quality IT Services BV

Dorothy Graham: On Testing
Dorothy Graham, Independent Consultant

Measurement and Metrics for Test Managers
Rick Craig, Software Quality Engineering

Innovation Thinking: Evolve and Expand Your Capabilities
Jennifer Bonine, tap\|QA, Inc.

Test Automation Strategies for the Agile World
Bob Galen, Velocity Partners

MAY 6–7

THE EXPO

Visit Top Industry Providers Offering the Latest in Testing Solutions

TOOLS
SERVICES
TECHNIQUES
DEMOS

WHO SHOULD ATTEND?

Software and test managers, QA managers and analysts, test practitioners and engineers, IT directors, CTOs, development managers, developers, and all managers and professionals who are interested in people, processes and technologies to test and evaluate software intensive systems

TO REGISTER CALL 888.268.8770 | STAREAST.TECHWELL.COM

Motivating Teams in a Self-Organizing Environment

By Maria Matarelli

THINKSTOCKPHOTOS.COM

Have you ever wondered how to motivate people in a self-organizing environment? Many companies are moving toward using an agile approach that fosters an environment of increased collaboration, cross-functional teams, and a completely different leadership style. In comparison to more traditional project management approaches, this leadership style is a fundamental shift from what we're used to. Rather than assigning and directing, we now need to guide, lead, and coach. But how do we get people to do things when we're not supposed to direct them on what to do? Imagine if you could tap into what truly motivates people.

Connecting with People

To allow people to step up, we have to step back and give them the room to take on more responsibility. Just like a tiger that lived in a six-foot cage its whole life, when you remove the cage, it may just walk back and forth a maximum of six feet, then lie down. The tiger was never allowed to go further. We need to let our team members know that it is okay to step up and lead; we need to encourage their self-organization. People don't always realize that they can begin to take more initiative, and for this to happen, we need to create an environment of safety to encourage people to try new things.

Rather than focusing on the process or metrics of what is completed, what if you really connected with the people on your teams and discovered what is truly important to them? As you explore the premises behind Relationship Awareness Theory, you will be able to empower your team members to better relate with others. Elias Porter, Ph.D., developed relationship awareness theory and created a reliable system for identifying people's motivational value systems using a quick and valid questionnaire known as Strength Deployment Inventory. [1] The Motivational Value Systems chart shown in table 1 summarizes these concepts as a foundation for better communication.

Motivational Value Systems	
	**Altruistic-Nurturing (BLUE)** Concern for the protection, growth and welfare of others
	**Assertive-Directing (RED)** Concern for task accomplishment and organization of resources to achieve results
	**Analytic–Autonomizing (GREEN)** Concern for well-thought out approaches, order, individualism, and self-reliance
	**Flexible-Cohering (HUB)** Concern for flexibility, group welfare, and team members
	**Assertive-Nurturing (RED-BLUE)** Concern for the protection, growth and welfare of others through task accomplishment and leadership
	**Judicious-Competing (RED-GREEN)** Concern for intelligent assertiveness, justice, leadership, order, and fairness in competition
	**Cautious-Supporting (BLUE-GREEN)** Concern for affirming and developing self-sufficiency in self and others, concern for thoughtful helpfulness with regard to justice

Table 1: Motivational value systems

The Benefit of Understanding What Motivates Your Team

Motivational value systems are a fairly consistent set of motives and values that influence people's perceptions and thoughts, what they focus on, and how they act.

We see the behavior of people we work with, even when the motivations and intentions behind their behavior are below the surface and not readily visible.

Every person has three primary motives: concern for people, performance, and process. These primary motives are represented by the colors blue, red, and green, respectively. Each person's motivational value systems are made up of a blend of these three primary motives. There are seven motivational value system types, based on the way the motives are blended.

Identifying the motivational value systems of the individuals on your teams can be a tool for understanding how to motivate and guide how they work together. You can tap into what's important to them rather than just directing what needs to be accomplished.

If someone on your team is primarily motivated by achievement, which is assertive-directing (red), you may want to talk about rallying the team around reaching the sprint goal and focusing on meeting the team's commitment.

If someone's motivational value system is altruistic-nurturing (blue), he may be more inspired by knowing how the customer will use the product and how many people it will serve and help. Bringing in a customer to provide feedback could help your team members see the end-user, which can be inspirational for team members motivated by an altruistic approach.

If a team member's motivational value system is analytic-autonomizing (green), you would want to share data and facts to identify the direction in which the team is moving and as a basis for decisions to be made.

While these are the three primary motivators, someone could be a blend of two or all three, so using a combination of these approaches to appeal to each of the team members' motivational values would be good.

Effective communication is a necessity for agile teams, yet a ScrumMaster may wonder, "How will my team know what to do if I don't tell them?" or "How do I get my team to self-organize?" If you can better understand the motivational value systems of the people on your team, you can appeal to them in a way that is compatible with their core motivational values to enhance communication and reduce conflict.

Applying Relationship Awareness Theory

There are four premises to relationship awareness theory:

Premise 1: Behavior is driven by motivation to achieve self-worth. Individuals are motivated to achieve this feeling of self-worth in different ways. Self worth can stem from being valued by yourself or by others. It is a universal motivation for people to want to feel a sense of self-worth. This is why it matters how we communicate with the people on our teams: sharing recognition, working together toward a common goal that aligns with a greater vision and mission, and recognizing the contributions of each person on the team.

Premise 2: Recognize that motivation changes in conflict as we aim to preserve our personal integrity and self-worth. Even though your primary motivation may be altruistic and nurturing (blue) when things are going well, that may not be the same way you are motivated when conflict arises. When conflict arises, you may become more analytical (green), and you may want to see data or facts to back up a decision or viewpoint. When conflict does arise, we may need to address people's motivational preferences before it becomes more of a problem.

Premise 3: When strengths are overdone or misapplied, they can be perceived as weaknesses. For example, confidence is a strength that someone on your team may have, but when it is overdone, the person may appear cocky or arrogant. Someone else may pay great attention to detail, but when overdone, it may come off as micromanaging other people on the team. Instead of thinking someone may have a bad quality that just detracts from how they interact with other team members, recognizing that it may just be an overdone strength can provide a basis for framing the conversation with that individual to see how that overdone strength could be dialed back.

Premise 4: Personal filters influence perceptions of ourselves and others. As we each have filters for how we see the world, our own personal filters influence how we judge others. People tend to use their own motivational values as a standard for how they expect other people to respond. When you look at the first premise of people's behavior being driven by motivation to achieve self-worth, it becomes easy to relate on a human level. By looking at a situation through their lens rather than your own, you can better understand and relate, which leads to more effective communication.

In a self-organizing team, if we don't understand each other and take other team members' perspectives into consideration, effective communication can be difficult.

After introducing these concepts to other team members while coaching agile teams in Chicago, it allowed us to better understand how to relate to each other. Several of my team members were altruistic and nurturing (blue) and trying to relate to others who were more assertive and directing (red). After seeing the differences in what motivated us as individuals when approaching how we worked together, it became apparent that we needed to appeal to other team members' primary motivations rather than just looking through our own lens. Even though several of us approached situations by focusing on the thoughts, feelings, and anticipated feedback of end-users, communicating with each other using dialogue that focused on the end goal and what needed to be accomplished helped to bridge the conversation gap. We wanted the same end result, but with different underlying motivations. This sometimes led to conflict or disagreements. We were able to reduce this conflict by better understanding each other's motivations and adjusting our language patterns accordingly.

When team members recognize the differences in other team members' core motivations that influence behavior, it allows an opportunity to adjust the way we interact with each other to better relate and, ultimately, improve how we communicate.

Hone Your Leadership Style

Traditional leadership embodies a command-and-control approach where it is an accepted practice to direct people on tasks and toward specific outcomes. As our industry moves toward leadership in an agile environment, it is necessary to adapt our leadership styles to encourage collaboration and self-organization among our teams. This includes increased focus on people and not just the tasks at hand.

When you are working with an agile team that is used to picking up one task after another without a cohesive goal, you may find yourself in a reactionary environment with fragmented work items being completed independent of one another. It is much more effective to share more of the vision and how your team's work aligns with the organization's goals. Identifying the direction the team should be heading will set the stage for what the team should focus on. After identifying the big-picture goals, applying relationship awareness theory and having an understanding of motivational value systems provide a comprehensive foundation for teams to work toward these larger goals.

While everyone on your team can benefit from this knowledge, it applies to any leader. Appealing to your team members' motivational values provides an alternative means to the command-and-control approach that may have been used before and offers insight into other ways to bring the team closer to communicating effectively.

How to Motivate Your Team

As we look at how to motivate team members in a self-organizing environment, it takes a different management style from what you may have used before.

Applying relationship awareness theory and motivational values systems to effectively tap into what motivates people can lead to higher productivity as well as increased engagement with your team.

In summary, it is important to connect with people and allow them room to step up. Recognizing others' motivational value systems allows us to enhance our communication and reduce conflict. Knowing the premises of relationship awareness theory provides great insights as you recognize others' desires to maintain their self-worth, remembering that motivations change when in conflict, and being aware of overdone strengths and applying your own filters.

When you adjust your leadership style to ensure your team members' unique motivational values are being considered while continuously sharing the overall vision, you will create an environment that naturally results in more effective communication, higher levels of collaboration, and enhanced self-organization. **{end}**

maria@formulaink.com

Click here to read more at StickyMinds.com.
- References


ENLIGHTENED
SOFTWARE TESTING
STRATEGIES
STAR
CANADA
JUNE 21–25, 2015
VANCOUVER, BC | CANADA
WESTIN BAYSHORE
REGISTER BY
APRIL 24, 2015
AND SAVE UP TO $300
GROUPS OF 3+ SAVE EVEN MORE
STARCANADA.TECHWELL.COM

IS AGILE BREAKING PRODUCT MANAGEMENT?

BY STEVE JOHNSON

THINKSTOCKPHOTOS.COM

What is the best way to organize the product owner role to produce the best results for leading an agile team? Some organizations say the role is primarily a business function; while others say it's primarily technical. Still others believe that the roles should be split, with the business role belonging to product marketing and the technical role belonging to product management.

A common definition of product management is the combination of product development and product marketing. Those in the agile community assume that product management represents both the business and the market to the product team. Some organizations use product management and product marketing to refer to the same function.

At a commercial software company, the product manager was writing product specifications, running beta programs, and managing schedules. She was too busy working closely with the development team to spend any time on the other product management activities, so new product releases were announced to the company like this: "The new build goes live tonight." (Not much of a launch, eh?) Chaos ensued as salespeople wanted to know how the marketing group was going to inform the customers about the new features before the next workday. As you might expect, no one in product marketing knew anything about the new release either, as they received notification at the same time as the salespeople.

A little marketing, a little business, a little technical support, plus project scheduling—that's today's product management. But what should a product manager focus on? And what about the product owner? Are these two roles or one? Getting clarity about roles is especially critical for agile projects due to the importance the business and customer perspective serves for the development team.

The Everything Role

Many technology companies assume the role of the product manager is to be the go-to resource for the product. Product managers are expected to be business and financial experts, product experts, domain and industry experts, and promotional experts.

The broad interpretation of product management simply covers too much. As a result, product managers are stretched way too thin.

I led a panel at an agile development conference where industry experts discussed the role of the product manager. I was shocked—and the audience was too—that none of the experts on the panel could agree on the responsibilities of the role. One panelist even pronounced, "The product manager's job is to do everything that nobody else wants to do."

Yikes! It is no wonder that product managers feel they have an impossible role—to be all things to all people, to be strategic, to be tactical, to be involved in both planning and execution, and to be a subject matter expert in every aspect of the product.

Product managers typically become experts based on previous job experience and as a result of their daily interactions with colleagues and customers.

Many organizations need a single person who is a subject matter expert on features, proposed future enhancements, what buyers and users think about the product, and the markets and industries served. Developers lacking knowledge about the market and users depend on a product manager or product owner to answer their questions. Marketing professionals lack knowledge about the product and its technology, so they expect a product owner to answer their questions. Finally, the sales staff relies on the product owner to answer their questions about the product and industry.

There is too much to do and not enough time.

The Transition of the Product Manager Role

Back in the 1980s, I joined a company as product manager. This was my first vendor experience, and it was one of the best-run software companies I have ever encountered—something I didn't appreciate until many years later. This company had a very clear job description for those who performed business planning for a single product. The title was simply called product manager.

In his seminal tech marketing book *Crossing the Chasm* [1], Geoffrey Moore recommended two separate titles that clearly distinguish the product manager from the product marketer:

"A product manager is responsible for ensuring that a product gets created, tested, and shipped on schedule and meets specifications. It is a highly internally focused job, bridging the marketing and development organizations and requiring a high degree of technical competence and project management experience.

A product marketing manager is responsible for bringing the product to the marketplace and to the distribution organization. It is a highly externally focused job."

In 1995, Ken Schwaber and Jeff Sutherland formalized the Scrum development methodology. [2] With it came yet another product management title: product owner.

"The product owner represents the stakeholders and is the voice of the customer. He or she is accountable for ensuring that the team delivers value to the business. Scrum teams should have one product owner."

What is the difference between a product manager and a product owner? And what about a product marketing manager?

The Skills Every Product Management Team Must Have

In recent years, we've seen new definitions for old titles and the creation of many new titles. We've got product managers, product marketing managers, product owners, business analysts, solution managers, product strategists, product line managers, portfolio managers, and more. Most agile teams have struggled with the differences, if any, between product manager and product owner. And what about the business analysts? Should they be product owners?

To keep it simple, there are four types of skills needed to define and deliver products to market. Product leaders need to

staff their teams with all four types of knowledge, but it's rare to find all of these capabilities in a single person.

Technology expertise: Technology expertise is about how the product works. From their daily interactions, technology experts pick up a deep understanding of a product's technical capabilities by working with the product and discussing it with customers and developers. For a technology expert, the product almost becomes a personal hobby. Typical titles that possess this expertise include the product owner, product manager, technical product manager, and business analyst.

Market expertise: Market expertise is a focus on geographic or vertical markets, either by country or by industry. A product manager who has market expertise knows how business is accomplished in that market. He knows the major players and the jargon of the market.

Market experts define themselves by the market they serve: "I'm a banker," or "I support BRIC." Typical titles with this expertise include the market owner, industry manager, product marketing manager, and field marketing manager.

Business expertise: Business expertise is where your traditional business leader or MBA graduate brings strength. These experts know the mechanics of business and can apply that knowledge to a product. A business-oriented expert knows how to use research to determine product feasibility and understands how the product generates profit, with financial analysis to back it up.

Ideally, these business skills need to be combined with one of the other skills or provided as a support role for the other areas of expertise. Typical titles with this knowledge include the business owner, product strategist, product leader, and portfolio manager.

Domain expertise: Domain expertise is about the discipline your product supports, such as security, fraud detection, or education. Domain experts are aware of (and sometimes even define) the standards for the discipline and can explain the latest thinking in that area. They understand the problems your product endeavors to solve, regardless of the market or industry.

Domain experts define themselves not by the product, but by their topic area, such as "I'm a security guy" or "My focus is on fundraising for nonprofits." Typical titles with this expertise include the product scientist and principal product manager.

You can see why product leaders struggle in some areas and breeze through others. Most of us understand these four product management skill sets and realize that it's difficult to find one person possessing them all, but we still typically look for one person when we should be seeking two or more.

Think about the skills you have and the skills you need for your organization. Consider the requests you're getting from development, marketing, sales, customer support, partners, and so on.

Determine which expertise is needed to accurately support your business. Then hire based on the skills you need.

Clarifying Titles and Roles with Agile Projects

Regardless of the agile methodology you're using, there are some critical roles that need to be performed.

The team: One of the great ideas found in agile methodologies is that an effective team is built of talented, clever people who bring many different skills. Rather than restrict what they can do, the agile team applies those skills as needed, without regard to title. You perform the job that needs doing.

This multifaceted team should have all the skills necessary to turn ideas into products, including design, programming, testing, and planning.

The process expert: The process expert or coach (called the ScrumMaster in Scrum) isn't a project scheduler—he's an expert and coach on the agile process you've chosen to use. The process expert is schooled in the methodology and leads the team through its mechanics, teaching techniques as required and enforcing the principles when necessary.

The process expert is the one who brings an egg timer to the daily stand-up to make sure the ten-minute checkpoint doesn't turn into an all-day meeting. At the end of each iteration, the process expert helps the team hone its implementation and alignment with the organization.

The product expert: Representing the customer in meetings and discussions is the role of the product expert (called the product owner in Scrum). Because this is typically an internally focused role, it is often a proxy for the business experts and market experts. The product expert helps the development team ensure that the features under development will be accepted by the customer.

Yet product experts can quickly become overwhelmed by the demands placed on them by the team and the rest of the company: to be the expert in the domain, the product, the market, and the business. If the product expert spends 100 percent of his time facilitating the development team, how can he keep up with the needs of the customer and the market? To be available at all times to the team, the product expert should work with only one team at a time.

The market expert: A market expert is externally focused, often gathering requirements from the market in order to define a product roadmap. The market expert focuses on sales enablement and go-to-market planning so that when the product is delivered, there are people who want to buy it.

The business expert: While the product expert is focused on supporting the development team and the market expert is focused on supporting the marketing and sales teams, a new role is needed to interact with the executive team: the business expert. The business expert maintains the portfolio roadmap and usually handles the business-related functions of the product—areas where the product expert and market expert are either overwhelmed or underskilled.

These four types of expertise may seem to overlap in some areas. When it is important ensure that roles and responsibilities are clearly understood, we need a way to resolve this.

Three Titles for Successful Product Management

Lately, I've been recommending some new titles for product managers. A product owner (sometimes called a technical product manager) should work closely with the development or engineering team; a market owner (sometimes called a product marketing manager) should be near the markets being served. Most of all, a business owner (or strategic product manager) should work closely with the business leaders and executives.

The job of the product owner is to keep development optimized with market and business information. Engineers don't want to be making business decisions—that's why they want instant access to a product owner. The product owner is best assigned to one product and to no more than two project teams.

A new role for many software development organizations is the market owner. This person understands the markets, segments, and personas. She should be the first point of contact for marketing and sales teams for product information. The market owner delivers prioritized market requirements to the business owner, who assigns them to the appropriate product.

The business owner is focused on the strategic aspects of the product within the portfolio. This can be somewhat political, requiring face time with the executive team due to the nature of the inquisition the business owner will undoubtedly face. What business are we in? What markets will we serve? Should we invest in this portfolio or that one? Should we retire the old stuff and invest in new stuff?

Ideally, the product owner and market owner should report to the business owner so all their objectives align. It's confusing to everyone when different vice presidents are providing different goals.

Agile Often Reveals How Fragile Other Systems Are

Agile methodologies optimize the product development process, but optimizing one part of a system often breaks other parts of the system. A typical agile adoption reveals suboptimal processes and understaffing in other groups. As a result, product management can't feed the process quickly enough, and sales and marketing teams cannot launch new products as quickly as they are being delivered. By fixing one constrained department, we reveal other constraints.

To succeed with agile, look beyond the development process to the activities that happen before development (like business planning and roadmapping) and after development (like launch and support). How will you get ideas into development in the right format? How will you support each step from idea to product to customer delivery? How will you show executives, colleagues, and customers where you're headed?

Product Management: A Key to High-Performance Agile Teams

Agile helps a high-performing team deliver more, better, faster. The idea is laudable: Members of the team bring their own skills and expertise and work together for a better tomorrow.

However, agile cannot make an effective team out of ambiguous roles and an ever-changing group of people. People who work together start building relationships. They come to appreciate different strengths and weaknesses. They begin by understanding how to work with different individuals—that one prefers a conversation while another prefers an email. Working relationships vanish every time the team composition changes. Here are my rules for success in an agile team.

Rule 1: Building products is definitely not factory work, although many leadership teams seem to think it is. Requirements do not come by on an assembly line, with developers coding them in a few minutes. It takes time for quality teams to deliver quality work.

Rule 2: Building products is creative, knowledge-based, highly skilled work and needs expert product guidance with clear roles.

Rule 3: The key to success is a team that has clear roles, works together, and stays together. An agile product management organization can help.

Yes, product managers should be agile! **{end}**

steve@under10consulting.com

Sticky Notes

Click here to read more at StickyMinds.com.

- References

Test Architect Introduces Integrated Test Automation Platform

TestArchitect test automation tools now include a fully integrated test automation environment and enhanced functionality that make it possible for organizations to achieve high levels of test automation for mobile, web, and desktop applications.

The integrated test automation platform combines three essential components: test development, automation technology, and test management. The single platform provides the ability to create highly reusable tests that can test applications across multiple operating systems. The automation technology can be easily customized to provide support for legacy as well as new applications as they are developed. The management component simplifies the end-to-end management of automation across the various stages of the application's lifecycle.

The tool's enhanced functionality includes the addition of Keypoint algorithm detection for sophisticated image recognition that makes it possible to automate testing for fast-moving gaming applications. A new optical character recognition (OCR) feature allows setting optimum OCR thresholds for images expected during testing. Support for iOS8 and Android 4.4, and sophisticated gestures makes it possible to test a wide range of Apple and Android mobile devices.

http://TestArchitect.com

API Readiness Platform for Secure APIs Now Available from SmartBear

SmartBear Software has announced that Ready! API, the industry's first fully integrated platform to help development, testing, and operations teams build reliable, scalable, and secure APIs, is available for download.

Ready! API is an "open core" API testing platform built on the foundation of the SoapUI open source tool and designed to provide all the tools needed when building and testing APIs.

Ready! API's service virtualization module, Service V Pro, includes virtualization capabilities, allowing developers to create virtual assets for parallel testing and development. All application modules available for Ready! API include:

- ServiceV Pro—for creating, managing, and sharing virtualized assets
- SoapUI NG Pro—the next generation of SoapUI for testing all web services, including REST and SOAP APIs
- LoadUI—for load testing of APIs to ensure they meet high performance demands
- Secure—for conducting security scans against APIs

http://smartbear.com/product/ready-api/free-trial

New Telerik Kendo UI Suite Solves Complex Data Manipulation Scenarios

Telerik has announced the latest updates to its HTML5 and JavaScript framework, Kendo UI suite. In the newest release, Kendo UI suite enables users to solve complex data manipulation scenarios more effectively, whether online, offline, or on-the-go.

Telerik is introducing two new data management widgets: TreeList, which combines grid and tree view features into one widget with load-on-demand and data editing capabilities; and the general availability of PivotGrid, the powerful data vizualization component that enables developers to perform operations over multidimensional/pivot data.

Telerik is releasing a new feature to the DataSource component, allowing apps to work in offline mode by storing data locally. Even when the connection stops, the application continues working and synchronizes changes immediately, once the connection is resumed.

With the newest release of Kendo UI suite, Telerik is adding client export capabilities for PDF/Excel. App users can now export data to PDF from all professional data management widgets, and to Excel from the grid.

http://telerik.com/kendo-ui

Keynote Unveils the Next Generation Performance Management Solution

Keynote has announced availability of its next generation performance management suite with new Real User Monitoring. This cloud-based solution empowers enterprises to optimize performance for the digital experience across web and mobile sites in less than three seconds. Keynote Real User Monitoring, built on big data architecture, combined with synthetic monitoring provides enterprises with real-time insight to drive revenue growth and enhance customer engagements.

Using Keynote's Performance Management Suite, now including Real User Monitoring, enterprises can determine the impact web slowdowns or errors have on user behavior in real time. This enables IT teams to identify possible solutions based on examining key metrics such as conversion rate, abandonment rate, and session duration.

This suite offers a flexible, open, and extensible platform that can be the basis of further analysis with business intelligence tools of your own choosing.

http://keynote.com/solutions/monitoring/web-monitoring

Crowdtesting Provider Testbirds Introduces Automated Testing Solution TestChameleon

Testbirds announced the arrival of TestChameleon, a new service that enables software testing with virtual machines (VMs). Through the service, companies and organizations are able to extensively test their applications on a wide range of operating systems, without the need of acquiring each environment for in-house testing. In essence, it facilitates thir running tests of their software to analyze for compatibility and inconsistencies.

TestChameleon is a cloud testing solution in which disposable VMs are constructed. The VMs are fully customizable, so clients get unrestricted administrator rights and can install any software for testing purposes—whether it's a single application or an entire .NET framework. Clients can securely run manual tests or automated tests with Selenium or SikuliX scripts.

http://testchameleon.com

Zeenyx Software Launches AscentialTest Version 7.2

Zeenyx Software, a testing solutions company, announced the release of AscentialTest version 7.2, which features integration with Jira for defect tracking and management. AscentialTest now provides a way for users to manage the relationship between tests, test results, and defects. Users can easily create defects and assign tests to defects from within their test repository.

By automatically keeping track of test results and defect status, AscentialTest prompts users to rerun tests to verify bug fixes and close defects based on test results.

V7.2 also provides automated testing support for out of band/asynchronous events.

http://zeenyx.com

Electric Cloud Launches Ship.io

Electric Cloud has announced it has publicly launched Ship.io to bring Continuous Delivery to native iOS and Android app development. Available as a free public beta service, Ship.io simplifies and automates build, test, and deployment processes for native iOS and Android apps.

As a SaaS offering, Ship.io gets mobile app developers building and testing their apps within minutes. Tightly integrated with GitHub, BitBucket, and other Git repositories, Ship.io automatically detects and configures native Android and iOS projects to allow teams to collaboratively build apps across supported OS, version, and device screen types. It connects existing tool chains to simplify and automate unit, functional and physical device testing, and allows teams to run these tests every time code is committed to provide fast feedback. Test frameworks in iOS and Android are supported. Ship.io builds also can be distributed to users through over-the-air (OTA) distribution, as well as through integrations with HockeyApp and TestFlight. Ship.io dashboards provide complete visibility into all build and test status levels, together with a central management platform for cross-team collaboration and historical reporting.

http://electric-cloud.com
https://ship.io

Xojo Announces Xojo 2014 Release 3

Xojo Inc. announces the general availability of Xojo 2014 Release 3, a major new release of their multi-platform rapid development tool that now includes the ability to develop native iOS applications.

With release 3, Xojo users can now build iOS apps in the same way desktop and web applications are created. Users can drag and drop to create the user interface and use one straightforward programming language to implement the functionality of their app.

Xojo 2014 Release 3 feature highlights include:

- New iOS framework
- New Text data type and supporting methods for improved text handling
- New Auto data type for storing any type or object
- New Xojo.Core classes that make creating international applications easier
- SSLSockets now support newer security protocols like TLS v1.1 and v1.2, and now defaults to TLS v1 instead of SSLv3
- For web applications, a new HandleURL event allows processing requests for URLs that previously could not be served. This feature also gives you the ability to respond to requests from Google and other search engines.

https://www.xojo.com/company/news/2014r3.php

Applause Launches Mobile Beta Management Offering

Applause, has launched Mobile Beta Management. While mobile beta tools typically constrain the number of beta users or level of engagement, Applause's Mobile Beta Management removes these limitations and enables companies to distill signal from noise to capture more actionable intelligence. The Applause Mobile Beta Management product further distinguishes itself by supporting global, enterprise-grade mobile beta programs with as many as hundreds of thousands of testers in a mobile-first framework.

Key features of Applause's Mobile Beta Management include:

- Mobile-first workflow, over-the-air distribution—users can download with a single tap
- Mobile NDA Management—enabling companies to protect IP and distribute quickly to users
- Automated session and crash reporting—providing insights into usage and app stability
- In-app feedback reporting—streamlining the entire beta experience
- The ability to solicit feedback from hundreds, thousands, or hundreds of thousands of your employees or early adopter users quickly and easily
- In-the-wild testing services—real-world testing for web, desktop, and mobile apps across the entire product lifecycle, including functional, usability, load, and localization testing with professional testers around the world

http://www.applause.com

Upcoming Events

April 12–17, 2015
San Diego, CA • USA
Manchester Grand Hyatt
mobiledevtest.techwell.com

May 3–8, 2015
Orlando, FL • USA
Gaylord Palms Resort
stareast.techwell.com

June 7–12, 2015
Las Vegas, NV • USA
Caesars Palace
adc-bsc-west.techwell.com

June 21–25, 2015
Vancouver, BC • Canada
Westin Bayshore
starcanada.techwell.com

September 27–October 2, 2015
Anaheim, CA • USA
Disneyland Hotel
starwest.techwell.com

November 2015
More Information Coming Soon!
adc-bsc-east.techwell.com

FAQ

expert answers to frequently asked questions

How Should You Prepare for the ISTQB Advanced Level Test Manager Certification Exam?

by Michael Sowers
msowers@sqe.com

The International Software Testing Qualification board (ISTQB), has defined several levels of certifications ranging from foundational to expert. By passing a comprehensive exam, a software testing professional can attain ISTQB's advanced-level certification as a test manager.

Unlike the three-day foundation-level certification, the advanced-level test manager class is focused on the role of the test manager. This is particularly important for anyone who has accountability for leading testing activities. The key topics are test process, test management, reviews, defect management, improving the test process, test tools and automation, and people skills. Extensive time is spent on the overall test process, test management, and people skills areas of the syllabus. The advanced-level test manager exam consists of sixty-five questions to be completed in three hours.

There is no magic formula in preparing for the exam, as each student's learning style varies. Based on feedback I've received from students and other instructors, this is a class where experience does matter. The more experience you have in a test manager role, the better.

Here are some recommendations in preparing for the exam.

1. ***Review the foundation-level material.*** This includes the syllabus, glossary, and any online or book material. The student must comprehend the foundation-level body of knowledge as a prerequisite to the advanced-level test manager certification.
2. ***Learn the syllabus and the glossary.*** Prior to attending an instructor-led class, study the complete advanced-level test manager syllabus and glossary. Pay close to attention to terms identified as "ATM."
3. ***During the instructor-led class, get on a learning schedule.*** Each night, review the syllabus up to the point of the daily instruction. Highlight the key areas from each section and subsection. Complete the practice exam questions for the particular material presented in the class each day, then review any of the supporting material. If you need additional clarification, discuss any questions with the instructor. Take your own notes. I like using mind maps to capture the key aspects and to organize relationships for each learning objective. Finally, complete the advanced-level test manager mock exam and review your answers with the supporting material.
4. ***Prepare for the final exam.*** Prior to taking the advanced-level test manager exam at the exam center, reread both the foundational and advanced syllabus and study the glossary terms marked with "F" and "ATM." Flip through the course notes set, review your class notes, and take additional practice exams.

To get the best results, customize this general guidance to the way you like to learn. The best advice I can offer is to allocate enough time to immerse yourself in the material through class discussions, guidance from the instructor, and disciplined self-study. **{end}**

Painful Lessons I Learned from Bootstrapping a Startup

Most of us have toyed with the fantasy of starting our own software company. Take to heart these seven lessons learned the hard way.

by Mike Botsko | botsko@gmail.com

In 2006, a designer friend and I joined forces to build a web-based client invoice management product. Even though we ran our own companies, we worked together on nearly every project. He was the designer and I was the developer.

Our product, Kiwi, needed to be completed quickly due to our own urgent demand for a better invoicing product. As soon as the app was working, we started using it ourselves, and soon after we launched a private alpha for a few local companies. Interest came quickly from clients, from health insurance agencies to manufacturers. We were happy with the early preview customer feedback.

Despite our enthusiasm, it wasn't long before other client work demanded too much of our time, and Kiwi was put on the back burner. Since there was no outside funding available to us, paid customer engagements dictated our priorities. I learned several valuable lessons from this situation.

> **"I sat down with my partner and told him I was unhappy with his lack of progress on our project. I needed him to resign."**

Lesson 1: Don't worry about losing focus on a side project. If you're still excited about a product's value, you'll be able to continue developing it when you have time again. It's the right project. After we cleared the queue of client work, I dusted Kiwi off and resumed work on the product.

Lesson 2: The moment you feel something isn't working, bring it up and get resolution. Don't settle until you get a resolution. Unsolved problems will only get worse. I began to notice a pattern with my partner: He was spending more time relaxing after hours than working. Initially, this wouldn't be something I'd worry about, but his ratio of time off to work sessions was shifting quickly.

It wasn't long before he was missing deadlines. I'd ask when tasks would be completed, only to hear empty promises. He was the most talented designer I'd ever worked with, and this was a work-for-equity project, so replacing him was likely impossible. I was powerless. I started feeling overworked and was tired of nagging. We considered seeking investment and gave a few preliminary pitches. My partner mentioned three friends who founded and sold a nautical software company. We decided that their business experience was just what the doctor ordered.

We brought all three into the company, and the five of us met weekly for a year. We planned the next phase of development, predicted financials, and talked about our exciting possibilities. But after a year of being the only one doing any actual work instead of just talking, the truth finally sunk in.

Lesson 3: Talkers are going to waste every resource you have. Hire the doers. Pride and excitement should grow from what you've completed, not what you hope will happen.

Although the three advisors had proven business experience, it became clear they weren't right for my business. They lost interest and became practically nonexistent for six months. By the time I had their approval to remove them from the company, they had essentially dropped off the face of the Earth.

Lesson 4: The business and the product is your first priority. Don't focus on anything that isn't contributing to that priority. At this point it was 2008 and the invoicing software market became oversaturated with competition. While we still felt our invoicing product was superior to anything in the market, we lost hope in attaining any market attention. Kiwi was officially abandoned and has since remained a private tool.

In October 2009, I realized that a better issue-tracking application was needed to attend to several problems we faced with client project work. I had long detested the poorly designed existing products and our clients refused to use them.

Lesson 5: Design your product for your own needs first. If you don't need the product or wouldn't use the product, why would anyone else?

I began development myself but quickly realized I needed a designer's magic touch. Timid from my experience with Kiwi, I asked my designer to again partner with me and produce designs for a new product called Snowy Evening. This time, we outlined some work agreements that would hopefully avoid past mistakes.

My partner's preliminary designs were amazing and motivated my development effort. By summer of 2010 the financial recession was hitting us both. Existing clients were unable to pay their bills—some went bankrupt, others canceled all work, and new clients were nowhere to be found. I spent most of my day marketing myself to potential clients and my nights on Snowy Evening. My partner did neither.

For several months, I'd approach him regularly, asking when he could resume work. He'd promise to start again, offering me a deadline—and missing it every time. At my wits' end, I began looking for a new partner. I assumed I was wasting my time because no one talented enough would be willing to work for equity. I was amazed to find a perfect candidate.

I sat down with my partner and told him I was unhappy with his lack of progress on our project. I needed him to resign. He was shocked and speechless. After thirty seconds, he started apologizing for everything. He opened up about personal issues that were deflating his motivation and outlined ways we could get back on track. He was clearly trying to save our friendship. I decided to give him a second chance.

Lesson 6: Sometimes, saving a friendship means stopping the partnership.

I allowed my emotions to finalize my decisions before I was able to sort out the problems. I was convinced the new candidate would come on board. However, the next day I was overwhelmed with a desire to make the current partnership work.

It was the biggest mistake I made with Snowy Evening. The new candidate understandably felt betrayed.

Due to financial problems, my partner couldn't afford his part of our office lease, so I had to work from home until some new work came in. Physical separation and lack of client work hurt our productivity, and Snowy Evening was again sidelined. My partner took a nontechnical design job that required him to be out of the city for months at a time.

Facing reality, I caught up with him and made it clear that his lack of commitment was unacceptable. Our partnership ended with his resignation. I became the sole custodian of Snowy Evening.

Lesson 7: Nothing motivates you like seeing people use your product.

I jumped back onto the project with a refreshed desire to complete Snowy Evening. I took over all unfinished work, removed all critical issues, and built a brand-new customer-facing sales page. By the fall of 2012, Snowy Evening was launched.

Snowy Evening has kept to its original roots: powerful enough for developers, yet amazingly usable for clients. I've begun an overhaul of the underlying technology. Real feedback from customers has been vital in helping me rewrite the code. With this new technical foundation, Ihave embarked on adding key features.

I'm extremely excited for the future growth of Snowy Evening. My design partner and I are still friends, and we regularly have lunch meetings to keep each other updated on work and life, but sadly, there's really nothing for us to work on together.

I no longer offer contracting services and am focused on building a successful product. While outside investments would have definitely made things easier, I'm glad I stuck to my dream and bootstrapped Snowy Evening's development. **{end}**

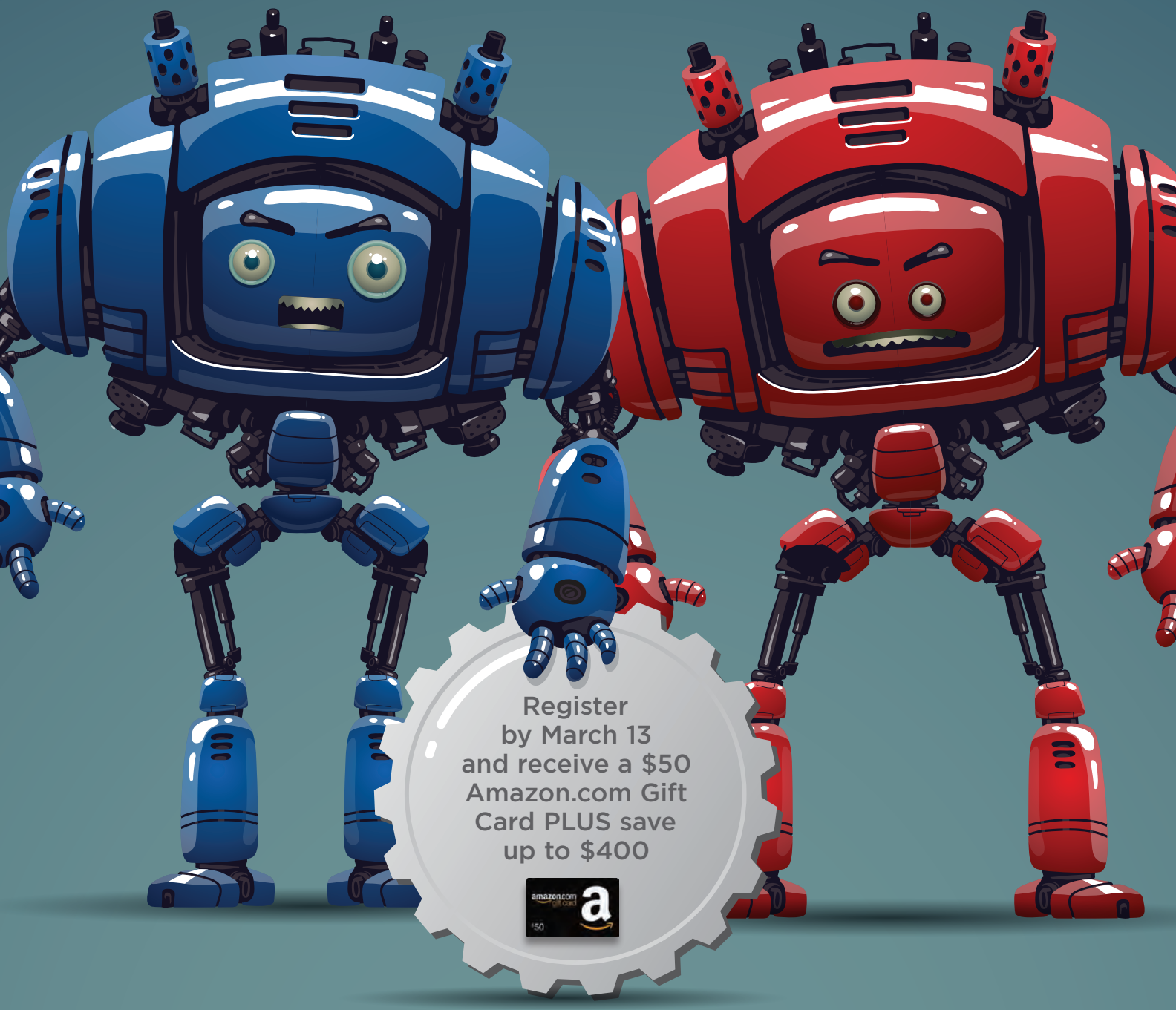
index to advertisers

Better Software (ISSN: 1553-1929) is published four times per year: January, April, June, and September. Print copies can be purchased from MagCloud (http://www.magcloud.com/user/bettersoftware).

AGILE DEVELOPMENT CONFERENCE WEST

LEAN MEAN SOFTWARE DEVELOPMENT MACHINE

BETTER SOFTWARE CONFERENCE WEST



Register by March 13 and receive a $50 Amazon.com Gift Card PLUS save up to $400

TWO CONFERENCES IN ONE LOCATION

REGISTER AND ATTEND SESSIONS FROM BOTH EVENTS!

June 7–12, 2015

LAS VEGAS, NV | CAESARS PALACE

Explore the Full Program at

ADC-BSC-WEST.TECHWELL.COM

R.E.P. Project Management Institute

PMI® members can earn PDUs at this event